

Remigiusz WIŚNIEWSKI, Łukasz STEFANOWICZ
 UNIwersYTET ZIELONOGÓRSKI, INSTYTUT INFORMATYKI I ELEKTRONIKI,
 ul. Licealna 9, 65-417 Zielona Góra

Zastosowanie hipergrafów w procesie selekcji implikantów prostych

Dr inż. Remigiusz WIŚNIEWSKI

Absolwent Uniwersytetu Zielonogórskiego, pracę doktorską obronił w 2008 roku. W latach 2000-2001 dwukrotnie odbył przemysłową praktykę studencką w firmie Aldec Inc. w Stanach Zjednoczonych. Aktualnie pracuje jako adiunkt (Uniwersytet Zielonogórski). Zainteresowania badawcze obejmują zagadnienia z zakresu teorii grafów i hipergrafów, bezpieczeństwa danych i kryptologii oraz metodologii projektowania i implementacji systemów cyfrowych z wykorzystaniem struktur programowalnych FPGA.



e-mail: R.Wisniewski@ie.uz.zgora.pl

Mgr inż. Łukasz STEFANOWICZ

Absolwent Uniwersytetu Zielonogórskiego, pracę magisterską obronił w 2011 roku. Jest słuchaczem studiów doktoranckich, specjalność informatyka. Członek PTI oraz Uczelnianego Koła Naukowego. Aktywnie uczestniczy w realizacji pokazów naukowych o zasięgu krajowym oraz międzynarodowym. Zainteresowania badawcze obejmują zagadnienia z zakresu teorii grafów oraz hipergrafów w kontekście sterowników logicznych.



e-mail: L.Stefanowicz@weit.uz.zgora.pl

Streszczenie

W referacie przedstawiona została nowa koncepcja selekcji implikantów prostych w procesie dwupoziomowej minimalizacji funkcji logicznych. Aktualnie znane metody selekcji bazują na połączeniu metod dokładnych z przybliżonymi. W artykule zaproponowana została nowatorska metoda selekcji, która w całości opiera się na algorytmach dokładnych, poprzez zastosowanie teorii hipergrafów. Najbardziej istotną zaletą proponowanego rozwiązania jest wielomianowa złożoność obliczeniowa całej operacji selekcji, która w przypadku ogólnym ma złożoność wykładniczą.

Słowa kluczowe: minimalizacja funkcji logicznych, selekcja implikantów prostych, hipergraf, transversala dokładna.

Application of hypergraphs to the prime implicant selection process

Abstract

In the paper a new idea for the selection of prime implicants is proposed. The method is based on the two-level minimization process of the Boolean functions, according to the Quine-McCluskey approach. Initially, the set of prime implicants for the logic function ought to be calculated. Next, the selection process is applied to achieve the minimal formula. Such an operation is a typical covering problem and in general case it has exponential computational complexity. In the paper we propose a new prime implicants selection method. An idea is based on the hypergraph theory. The prime implicants table is formed as a selection hypergraph. If the selection hypergraph belongs to the Exact Transversal Hypergraph class (xt-class), the solution may be obtained in a polynomial time, which is not possible in a general case. The proposed method is illustrated by an example. All necessary steps are shown in order to apply the proposed selection algorithm to minimize an exemplary Boolean function.

Keywords: minimization of Boolean functions, selection of prime implicants, hypergraph, exact transversal.

1. Wprowadzenie

Minimalizacja funkcji logicznych to jedna z podstawowych operacji algebry Boole'a, wykorzystywana podczas projektowania systemów cyfrowych w celu zmniejszenia kosztów produkcji poprzez usunięcie zbędnych bramek [1, 2, 3]. Choć powszechnie używana jest nazwa „minimalizacja funkcji”, faktycznie redukcji ulega formuła opisująca daną funkcję logiczną. Zastosowanie minimalizacji jest szczególnie istotne w przypadku układów logicznych [4, 5, 6]. Uzyskanie jak najbardziej zredukowanej postaci pierwotnej funkcji logicznej ma kluczowe znaczenie w różnego rodzaju procesach związanych z implementacją systemu w docelowym układzie cyfrowym, zarówno w przypadku systemów dedykowanych, jak i układów programowalnych [1, 7, 8, 9].

Tradycyjne i najbardziej znane algorytmy, bazują na wykorzystaniu dwupoziomowej metody minimalizacji. W pierwszej fazie wyznaczane są implikanty proste funkcji logicznej, natomiast drugi etap to ich selekcja. Tak funkcjonuje graficzna metoda

siatek (map) Karnaugh'a, oraz bardziej zaawansowany algorytm Quine'a-McCluskey'a (QMC) [2, 10, 11].

Pomimo, że metody te powstały ponad pół wieku temu, do dziś są powszechnie stosowane, a wszelkie nowatorskie pomysły w większości stanowią rozwinięcie lub częściowe ulepszenie dwupoziomowego algorytmu minimalizacji. Podstawowym problemem, zarówno podczas wyznaczania, jak i selekcji implikantów prostych, jest wykładnicza złożoność obliczeniowa tych operacji [3]. Dlatego też ośrodki naukowe oraz firmy komercyjne intensywnie poszukują efektywnych metod, usprawniających cały proces minimalizacji. Szczególnie popularne są algorytmy algebraiczne oraz genetyczne [12, 13, 14].

2. Sformułowanie problemu

Pomimo wielu odmian, generalna zasada wszystkich metod dwupoziomowej minimalizacji funkcji logicznych jest podobna. Pierwszy krok to wyznaczenie zbioru implikantów prostych, które są najczęściej zestawiane w formie tablicy mintermów oraz pokrywających je implikantów prostych. Selekcja sprowadza się do znalezienia pokrycia jak najmniejszą liczbą implikantów wszystkich mintermów funkcji. Proces ten można zasadniczo podzielić na dwa etapy. Pierwszy z nich to redukcja tablicy pokrycia metodą dokładną. Jest to tzw. algorytm *Cyclic Core* (w polskiej literaturze nazwany także *Szybkim Algorytmem Redukcji* [1], w niniejszym artykule stosowana będzie oryginalna nazwa), pierwotnie zaimplementowany jako część systemu *ESPRESSO* [15]. Jest to algorytm dokładny i wiąże się z wielomianową złożonością obliczeniową. Druga faza, to właściwa selekcja implikantów prostych na podstawie zredukowanej tablicy pokrycia. Etap ten w ogólnym przypadku jest wykładniczy. Oznacza to, że rozwiązanie dokładne może nie zostać znalezione w zakładanym czasie (lub nawet w ogóle nie zostanie znalezione). Dlatego też stosowane są metody przybliżone (np. algorytmy zachłanne, genetyczne, lub inne rozwiązujące problem pokrycia heurystycznie). Takie podejście oznacza, że wynik zostanie znaleziony w czasie wielomianowym, natomiast nie gwarantuje optymalnego rozwiązania (np. może się okazać, że liczba implikantów prostych jest nadmiarowa).

W literaturze można znaleźć metody usprawniające proces selekcji implikantów prostych. Przykładowo koncepcja zaproponowana w [14] jest abstrakcyjnym rozwinięciem algorytmu pokrycia dokładnego *Cyclic Core*, w której wykorzystano tzw. funkcje transpozycji. Problem zostaje sprowadzony do pokrycia zbioru $\langle X, Y \rangle$, w którym zarówno X , jak i Y są podzbiorem kraty (Z, \sqsubseteq) , natomiast y jest pokryciem x wtedy i tylko wtedy, gdy $x \sqsubseteq y$. Dzięki temu selekcja może zostać przeprowadzona za pomocą algorytmu *Cyclic Core* oraz z zastosowaniem grafów *BDD* [14].

Oprócz funkcji transpozycji należy zwrócić uwagę na redukcję *Gimpla* [14], która de facto jest także rozwinięciem metody *Cyclic Core*. Co prawda umożliwia ona redukcję kolumn, jednak w najgorszym przypadku wprowadza dodatkowe wiersze, co sprowadza się do utworzenia nowych dominujących wierszy oraz kolumn.

W praktyce redukcja Gimpla ma sens jedynie w przypadku gwarancji uzyskania mniejszej macierzy pokrycia.

W niniejszym artykule proponujemy innowacyjne podejście w kwestii właściwej selekcji implikantów prostych. Uzyskana tablica pokrycia jest obrazowana za pomocą hipergrafu selekcji. Następnie, weryfikowana jest klasa hipergrafu selekcji. Jeśli należy on do klasy hipergrafów typu xt [16], to oznacza, że cały proces selekcji może zostać przeprowadzony w czasie wielomianowym. W takim przypadku problem pokrycia sprowadza się do znalezienia pierwszej transwersali dokładnej i wiąże się z wielomianową złożonością obliczeniową [16, 17].

W referacie ograniczono się do gruntownej analizy drugiego z etapów minimalizacji funkcji logicznych. Przyjęto, że w pierwszym etapie wyznaczone zostały wszystkie implikanty proste dla danej funkcji, a celem jest sprawdzenie, czy dokładny proces selekcji może zostać wykonany w czasie wielomianowym. Oczywiście, nawet jeśli selekcja będzie miała złożoność wielomianową, to sam proces wyznaczania implikantów prostych może być wykładniczy. Ten aspekt jest jednakże poruszany w wielu publikacjach [2, 3, 12], gdzie proponowane są różne sposoby usprawniające operację wyznaczania implikantów prostych (metody heurystyczne, także takie, które generują ograniczony zbiór implikantów prostych).

3. Podstawowe definicje

Poniżej zamieszczono podstawowe definicje, związane z teorią hipergrafów, które są stosowane w niniejszym artykule [5, 16, 17, 18]:

Hipergraf H definiuje dwójka:

$$H = (V, E) \quad (1)$$

gdzie: $V = \{v_1, \dots, v_n\}$, jest dowolnym, niepustym zbiorem wierzchołków; $E = \{E_1, \dots, E_m\}$, jest zbiorem krawędzi hipergrafu, czyli podzbiorem zbioru $P(V)$ wszystkich możliwych niepustych zbiorów, których elementy należą do V .

Transwersalą (pokryciem wierzchołkowym) hipergrafu H jest zbiór $T \subset V$ zawierający wierzchołki incydentne do każdej krawędzi hipergrafu:

$$T \cap e_i \neq \emptyset \quad (i = 1, \dots, m) \quad (2)$$

Transwersala minimalna to transwersala, która nie zawiera żadnej innej transwersali hipergrafu H .

Transwersalą dokładną hipergrafu H jest zbiór $D \subset V$ wierzchołków hipergrafu, przy czym każda krawędź hipergrafu musi być incydentna z dokładnie jednym wierzchołkiem, wchodzącym w skład transwersali dokładnej.

Hipergraf transwersal dokładnych (ang. *exact transversals hypergraph*, *xt-hypergraph*) to hipergraf, w którym wszystkie minimalne transwersale są jednocześnie transwersalami dokładnymi. Hipergraf H należy do **klasy xt** , jeśli jest hipergrafem transwersal dokładnych. Hipergrafy transwersal dokładnych mają bardzo charakterystyczne własności. Niewątpliwie najistotniejszą zaletą jest możliwość wyznaczenia pierwszej oraz kolejnych transwersal dokładnych w czasie wielomianowym [16]. Ponadto, sprawdzenie, czy dany hipergraf należy do klasy xt może być także wykonane wielomianowo [16].

4. Idea proponowanej metody

Proponowana metoda selekcji implikantów prostych wykorzystuje własności hipergrafów oraz transwersal dokładnych. Pierwszy etap jest taki sam jak w podejściu klasycznym; wykonywana jest operacja redukcji tablicy pokrycia (*Cyclic Core*). Następnie, uzyskane rezultaty zestawiane są w formie hipergrafu selekcji.

Jego wierzchołki odpowiadają implikantom prostym, natomiast krawędzie obrazują mintermy funkcji. W dalszej fazie, weryfikowana jest klasa hipergrafu. Jeśli jest on hipergrafem transwersal dokładnych, to oznacza, że cały proces selekcji może zostać wykonany wielomianowo. Wówczas obliczana jest pierwsza transwersala dokładna, która jednoznacznie wskazuje szukane implikanty proste. Jeśli jednak okaże się, że hipergraf selekcji nie należy do klasy xt , wówczas proponowana metoda nie może zostać zastosowana i rozwiązanie zostaje znalezione z wykorzystaniem metod przybliżonych. Koncepcja proponowanej metody została przedstawiona na rys. 1. W ramce oznaczono najistotniejsze operacje proponowanego rozwiązania.

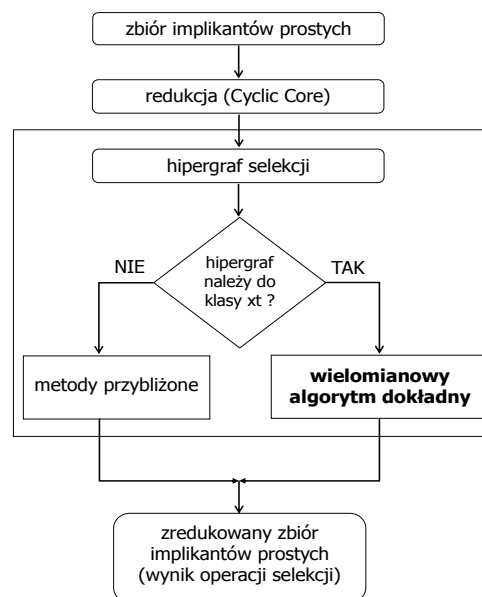


Fig. 1. Flow-chart showing an idea of the proposed method
Rys. 1. Schemat ilustrujący ideę proponowanej metody selekcji

Przedstawiona metoda selekcji implikantów prostych może zostać podzielona na następujące fazy:

1. **Redukcja tablicy pokrycia metodą dokładną.** Etap ten jest wykonywany zgodnie z procedurą opisaną w [15].
2. **Określenie hipergrafu selekcji H_S .** W tym kroku implikanty proste zredukowanej tablicy zostają zobrazowane za pomocą wierzchołków, natomiast mintermy za pomocą krawędzi hipergrafu selekcji H_S .
3. **Sprawdzenie, czy hipergraf selekcji jest hipergrafem transwersal dokładnych.** Test wykonywany jest zgodnie z procedurą przedstawioną w [16].
4. **Jeśli H_S jest hipergrafem transwersal dokładnych,** szukana jest pierwszą najmniejsza transwersala dokładna. Proces ten ma wielomianową złożoność obliczeniową [16, 17]. Efektywną implementację algorytmu pokazano w [19] oraz [17].
5. **Jeśli H_S nie należy do klasy xt ,** wówczas poszukiwane jest rozwiązanie przybliżone. W szczególności może to być na przykład znalezienie transwersali T w hipergrafie H_S . Istnieje możliwość zastosowania algorytmów dokładnych, jednakże ich złożoność obliczeniowa jest wykładnicza [18, 19].
6. **Szukane rozwiązanie stanowi zbiór implikantów prostych,** określonych na podstawie transwersali hipergrafu H_S obliczonej w punkcie 4 lub 5.

5. Przykład zastosowania omawianej metody

Przedstawiona metoda selekcji zostanie zilustrowana przykładem. Załóżmy, że procesowi minimalizacji poddana zostaje następująca funkcja logiczna:

$$f(x_1, x_2, x_3, x_4) = \Sigma(0, 5, 10, 15) + d(1, 2, 4, 7, 8, 11, 13, 14) \quad (3)$$

W powyższej funkcji wyróżnić można cztery mintermy należące do zbioru włączenia oraz osiem, w którym funkcja jest nieokreślona [5, 6]. Pierwszym etapem minimalizacji jest wyznaczenie implikantów prostych. Dla danego przykładu zbiór implikantów prostych składa się z czterech wartości (w nawiasach podano notację obrazującą kostki, uzyskane w pierwszym etapie minimalizacji [5]):

$$\bullet I_1 = \overline{x_1} \overline{x_3} (0*0*),$$

$$\bullet I_2 = \overline{x_2} \overline{x_4} (*0*0),$$

$$\bullet I_3 = x_2 x_4 (*1*1),$$

$$\bullet I_4 = x_1 x_3 (1*1*).$$

Tablica pokrycia dla rozpatrywanej funkcji została przedstawiona za pomocą tab. 1. W celu zwiększenia przejrzystości, w nawiasach zamieszczono binarne wartości mintermów oraz implikantów.

Kolumny tablicy reprezentują implikanty proste, natomiast wiersze – mintermy funkcji. Jeśli dany minterm jest pokrywany przez implikant, w tabeli oznaczane jest to wartością „1”, w przeciwnym wypadku do tablicy wpisywana jest wartość „0”.

Tab. 1. Tablica pokrycia dla analizowanego przykładu
Tab. 1. Cover table for the example

Mintermy funkcji	Implikanty proste			
	$I_1 (0*0*)$	$I_2 (*0*0)$	$I_3 (*1*1)$	$I_4 (1*1*)$
0 (0000)	1	1	0	0
5 (0101)	1	0	1	0
10 (1010)	0	1	0	1
15 (1111)	0	0	1	1

Pierwszym etapem proponowanej metody selekcji jest redukcja tablicy metodą dokładną. Zgodnie z procedurą, przedstawioną w [15], z tablicy usuwane są dominujące kolumny oraz zdominowane wiersze. W przedstawionym przykładzie operacja ta nie przyniesie żadnych zmian, ponieważ żadna kolumna nie dominuje nad inną, ponadto nie można wyróżnić zdominowanych wierszy. Dlatego też hipergraf selekcji H_S zawiera cztery wierzchołki, które odpowiadają implikantom prostym, oraz cztery hiperkrawędzie, które z kolei obrazują mintermy funkcji.

Ponieważ hipergraf H_S należy do klasy xt, możliwe jest określenie rozwiązania w czasie wielomianowym. W tym celu obliczana jest pierwsza transwersała dokładna hipergrafu H_S . W przedstawionym przykładzie będzie ona zawierać dwa elementy: $D = \{I_1, I_4\}$. Oznacza to, że w procesie selekcji określone zostały dwa implikanty proste. Na tej podstawie określana jest formuła minimalna dla rozpatrywanej funkcji:

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \overline{x_3} + x_1 x_3 \quad (4)$$

6. Podsumowanie

W artykule zaprezentowano innowacyjny algorytm selekcji implikantów prostych w procesie minimalizacji funkcji logicznych.

Przedstawiona koncepcja bazuje na dwupoziomowej metodzie minimalizacji, w której najpierw określone są implikanty proste. Wynik tej operacji jest następnie obrazowany za pomocą hipergrafu, którego wierzchołki odpowiadają wyznaczonym implikantom prostym, zaś krawędzie mintermom funkcji. Po wstępnej redukcji określany jest typ hipergrafu selekcji. Jeśli należy on do klasy xt, wówczas cały proces selekcji jest przeprowadzany wielomianowo. W przeciwnym wypadku zastosowanie mają klasyczne metody wyznaczania transwersali hipergrafu. Celem proponowanej metody jest redukcja złożoności obliczeniowej, jeśli hipergraf selekcji należy do klasy xt. Sprawdzenie typu hipergrafu oraz wyznaczenie transwersali dokładnej w hipergrafie xt wiąże się z wielomianową złożonością obliczeniową, natomiast w przypadku ogólnym jest to problem wykładniczy.

7. Literatura

- [1] De Micheli G.: Synthesis and Optimization of Digital Circuits, McGraw-Hill, New York, NY, 1994.
- [2] Coudert O.: Two-Level Logic Minimization: An Overview, Integration Vol. 17, No. 2, pp. 97-140, Oct. 1994.
- [3] Coudert O., Madre J. C., Fraise H.: A New Viewpoint on Two-Level Logic Minimization, Proc. 30th Design Automation Conference, Dallas, TX, USA, pp. 625-630, June 1993.
- [4] Maxfield C.: The Design Warrior's Guide to FPGAs, Academic Press, Inc., Orlando, FL, 2004.
- [5] Łuba T.: Synteza układów cyfrowych, WKŁ, Warszawa, 2003.
- [6] Kania D.: Układy logiki programowalnej, PWN, Warszawa, 2012.
- [7] Barkalov A., Titarenko L.: Logic synthesis for FSM-based control units, Lecture Notes in Electrical Engineering, Vol. 53. Springer-Verlag, Berlin, 2009.
- [8] Grobelna I.: Formal verification of embedded logic controller specification with computer deduction in temporal logic, Przegląd Elektrotechniczny, 12a, 40-43, 2011.
- [9] Milik A., Hryniewicz E.: Synthesis and implementation of reconfigurable PLC on FPGA platform”, International Journal of Electronics and Telecommunications, 58 (1), 85-94, 2012.
- [10] Quine W.V.O.: The problem of simplifying truth functions, American Math. Monthly, Vol. 59, pp. 521-531, 1952.
- [11] McCluskey E. J.: Introduction to the Theory of Switching Circuits, McGraw-Hill, New York, 1965.
- [12] Nosrati M., Karimi R., Hariri M.: Minimization of Boolean Functions Using Genetic Algorithm, Annals. Computer Science Series, Vol. X, 2012.
- [13] Nosrati M., Hariri M.: An Algorithm for Minimizing of Boolean Functions Based on Graph DS, World Applied Programming Vol. 1 No.3, p. 209-214, 2011
- [14] Coudert O., Sasao T.: Two-Level Logic Minimization, Logic Synthesis and Verification, S. Hassoun & T. Sasao Editors, Kluwer Academic Publishers, Chapter 1, pp. 167-196, 2001.
- [15] Rudell R. L.: Logic Synthesis for VLSI Design. Rozprawa doktorska, EECS Department, University of California, Berkeley, 1989.
- [16] Eiter T.: Exact Transversal Hypergraphs and Application to Boolean μ -Functions. Journal of Symbolic Computation, Vol. 17, No. 3, pp. 215-225, 1994.
- [17] Wiśniewska M.: Application of hypergraphs to the decomposition of the discrete systems, Lecture Notes in Control and Computer Science, Vol. 23. Univ. of Zielona Góra Press, Zielona Góra, 2012.
- [18] Berge C.: Graphs and Hypergraph. Amsterdam: North-Hols.r Mathematical Library, 1976.
- [19] Knuth D.: Dancing Links, Stanford University, 2000, <http://www-cs-faculty.stanford.edu/~uno/papers/dancing-color.ps.gz>
- [20] Adamski M., Karatkievich A., Węgrzyn M. (ed.): Design of Embedded Control Systems. NY, Springer Science, 2005.